Universitat de Lleida

TREBALL FINAL DE MÀSTER

Estudiant: **Rossend Rocamora Redondo**

Titulació: Màster en Enginyeria Informàtica

Títol de Treball Final de Màster: Disseny i implementació d'una aplicació web per al registre de fitxers utilitzant tecnologia blockchain

Director/a: Josep Argelich Romà i Josep Maria Miret Biosca

Presentació

Mes: Juliol

Any: 2019

Disseny i implementació d'una aplicació web per al registre de fitxers utilitzant tecnologia blockchain

Autor: Rossend Rocamora Redondo

Directors: Josep Argelich Romà i Josep Maria Miret Biosca

Universitat de Lleida
Escola Politècnica Superior
Màster en Enginyeria Informàtica
Treball de Final de Màster

9 de juny de 2019

Índex

Índex **3**

Índex de figures **5**

1 Introducció **11**
1.1 Antecedents i motivació 12
1.2 Propòsit i estructura d'aquest treball 12

2 Conceptes bàsics de Criptografia **15**
2.1 Història 15
2.2 Algoritmes d'autentifiació 18
2.3 Algoritmes criptogràfics 19
2.3.1 Intercanvi de claus Diffie-Hellman 19
2.3.2 Criptografia asimètrica o de clau pública 20
2.3.3 Criptografia amb corbes el·líptiques 21

3 Fonaments del Bitcoin i de la Blockchain **23**
3.1 Protocol P2P 24
3.2 Blockchain 25
3.2.1 Conceptes màtematics de la blockchain 28
3.2.2 Prova d'existència sobre blockchain 29

4 Implementació i prova de concepte(d'existència) **31**
4.1 Creació d'un node Blockchain 31
4.2 Treballar amb la Blockchain 34
4.3 Creació d'una Webapp de prova d'existència 37

5 Conclusions **41**

Bibliografia **43**

Índex de figures

2.1 Matriu de Polybios 17
2.2 Xifrat en cèsar 17
2.3 Codificador enigma 18
2.4 Funció resum (hash function) 19
2.5 Taula comparativa entre longitud/seguretat entre RSA, ECDLP i AES 22

3.1 Protocols d'Internet 23
3.2 Blockchain 27

4.1 Nodes a nivell mundial 32
4.2 Node local 33
4.3 Transacció a blockchain.info 36
4.4 Esquema de funcionament del WebApp 37
4.5 Pas 1: Pagina Principal de la nostra WebApp 38
4.6 Pas 2: Apartat de la nostra WebApp que ens permet seleccionar el fitxer sobre el que volem crear o visualitzar marca de temps 38
4.7 Pas 3: Una vegada tenim el fitxer seleccionat, apliquem el check de validació de RecaptCha 39
4.8 Pas 4: En aquest punt l'aplicació extreu el MD5 del fitxer pujat i el puja a la Blockchain de BTC amb un import molt petit i amb el MD5 del fitxer en el camp OP RETURN 39
4.9 Pas 5: En aquesta captura observem que la WebApp retorna el numero de transacció en la Blockchain de BTC. Un enllaç per a que vuguem veure la nostra transacció en blockchain.com, la marca de temps i el MD5 del Fitxer. 40

Agraïments

Primerament donar gracies al Josep Maria Miret i al Josep Argelich per la seva paciència en la realització d'aquest treball. També agrair a l'Albert Mas per introduir-me en tot el mon de les criptodivises. Gracies al Nestor Torres per l'ajuda oferia en la generació de la interfície de la WebApp i per continuar treballant en la idea base del meu projecte fins a convertir-ho en una eina empresarial https://guardo.vunkers.com/.

Resum

La finalitat d'aquest Treball de Final de Màster és la creació d'una prova d'existència de fitxers basada en la Blockchain de Bitcoin. Així doncs, es composa d'una petita aplicació en PHP que digereix un fitxer pujat per extreure'n un identificador únic i el puja a la Blockchain de Bitcoin. En cas de pujar-lo per segona vegada el sistema ens retorna l'escriptura a Blockchain amb la marca de temps que correspon al fitxer.

Per a la realització d'aquest treball hem seleccionat el llenguatge de programació PHP ja que hem trobat un gran nombre de llibreries PHP preparades per atacar a les API's de la Blockchain. Per altra banda ha estat necessària la instal·lació i configuració d'un node de Blockchain per a poder realitzar escriptures usant un parametre especial que ens permet escriure un nombre determinat de bits en un camp lliure en cada transacció. Al llarg d'aquest treball s'han emprat diversos algoritmes de digestió així com diferents tipus de API's per assolir l'objectiu.

Per a realitzar aquest treball de final de màster he hagut d'assolir una sèrie de coneixements sobre criptografia, algoritmes de digestió i principalment sobre el funcionament de la Blockhain de Bitcoin.

Capítol 1

Introducció

Al llarg de la història, un dels grans reptes que aparegué arran de la comunicació no verbal, fou intentar esbrinar el remitent o garantir la veracitat de les dades que eren rebudes. En el moment que deixem de tenir visible el nostre interlocutor aquest deixa de ser per a nosaltres un remitent segur. Durant la història doncs, s'han proposat una sèrie de mesures per a verificar el remitent i moltes ens acompanyen fins avui. Els títols universitaris, hipoteques, fins i tot algunes targetes de crèdit basen la credibilitat del remitent en una signatura manual. Aquesta signatura és normalment fàcilment falsificable per a l'ull no expert fins al punt que existeix un col·lectiu que únicament es dedica a verificar signatures manuals.

Ja ben entrats en l'era digital, moltes empreses han vist en la verificació del remitent o la integritat de les dades un nínxol de mercat que han intentat explotar amb més o menys èxit. Principalment les podríem classificar en:

1. Matemàticament: Algunes empreses han aprofitat tots els avenços en l'àmbit criptogràfic per a crear productes robustos sobre fonaments matemàtics, concretament en models objectius. Moltes vegades aquests avenços, tot i ser irrebatibles, mai ofereixen garanties del 100% ja que matemàticament mai es podrà dir que una clau és IMPOSSIBLE de desxifrar.

2. Legalment: Algunes altres empreses s'han cenyit als valors legals invertint una gran quantitat de diners en ISOS: certificacions i jocs de proves per legalitzar la seva forma de verificació o certificació. L'exemple més exagerat és la banca: els sistemes informàtics d'aquesta segueixen funcionant amb bases de dades, relacions i processos d'importació fora

de les hores productives. La lògica és aplicada per software de gestió d'aquestes bases de dades.

En aquesta direcció en Satoshi Nakamoto va optar per aplicar paràmetres matemàtics i de criptografia per a crear un sistema de Base de Dades distribuït que inclogués implícitament un sistema de gestió monetari. El sistema va ser anomenat Blockchain i ha estat tota una revolució. En el capítol 3 veurem en profunditat en que consisteix la primera prova del concepte que a dia d'avui segueix sent funcional.
Aquest concepte de base de dades orientada a tractar transaccions monetàries està servint a part de crear un nou sistema monetari basat en la robustesa critpogràfica, també per a crear tota una sèrie de noves formes d'emmagatzematge de dades.

1.1 Antecedents i motivació

Els antecedents d'aquest projecte es remunten a l'any 2009: els orígens de la criptodivisa Bitcoin. Pocs mesos després ens va arribar a les mans un client amb una necessitat un tant especial, havia set atacat per una de les primeres versions de Cryptolocker (el software de moda que xifra les dades dels servidor). El segrestador demanava un rescat de 2000 BTC i ningú sabia, ni tenia massa clar com aconseguir-los i menys encara realitzar el pagament. Així doncs es vam posar a investigar com generar els nostres propis Bitcoins i vam poder solucionar el problema. En aquell punt em va quedar clar que aquesta nova forma tenia un potencial increïble i vaig voler saber com treballava el sistema. Així vaig descobrir Blockchain, el propi concepte obria tot un món de possibilitats.

1.2 Propòsit i estructura d'aquest treball

El propòsit d'aquest treball és donar una aportació pràctica a Blockchain en forma de prova d'existència. Definirem prova d'existència d'un fitxer com una forma de demostrar que un document digital de qualsevol tipus i format va existir en les mateixes condicions en una data concreta que quedarà enregistrada en la pròpia Blockchain.

Aquest treball s'estructura de la següent manera. Començarem parlant de Criptografia, ja que tots els fonaments del treball estan basats sobre Criptografia. Tot seguit introduirem el concepte de Blockchain i Bitcoin tot resseguint la seva curta historia i característiques. Per anar acabant

es presentarà un ús pràctic de la Blockchain de Bitcoin presentant una implementació de la prova d'existència sobre la Blockchain de Bitcoin. Finalment parlarem de les conclusions extretes del treball.

Capítol 2

Conceptes bàsics de Criptografia

> "La criptografia [1] (o criptologia) és, tradicionalment, l'estudi de formes de convertir informació des de la seva forma original cap a un codi incomprensible, de forma que sigui incomprensible pels que no coneguin aquesta tècnica. La criptografia moderna utilitza les disciplines de les matemàtiques, la informàtica i l'electrotècnia."

En aquest capítol, presentarem la criptografia, resseguirem la historia de la criptografia entrarem en detall en els conceptes criptogràfics que ens són d'interès per aquest treball.

2.1 Història

Des dels inicis de la humanitat, l'home ha tingut la necessitat de comunicar-se amb la resta d'éssers vius a través de senyals, gestos, sons i sorolls. Temps després va començar a transferir-los com a mitjà gràfic.

Aquesta comunicació va sorgir per la necessitat de donar informació a la resta d'humans. Al principi, els éssers humans es comunicaven a través de gestos i símbols que feien amb el seu cos i temps després van incloure els sons per a poder simplificar i fer més fàcil aquesta comunicació entre ells. S'atribuïa un soroll o so a cada cosa que els envoltava i poc a poc van anar realitzant el seu llenguatge fins a poder arribar a un llenguatge més desenvolupat.

Un cop els éssers humans van anar adquirint coneixements més variats, complexos i més informació respecte el món que els rodejava es van anar civilitzant en un conjunt de persones amb una millor evolució i desenvolupament del seu estil de vida i del seu raonament i forma de comunicar-se.

Una de les primeres civilitzacions, l'antic Egipte, va tenir una millora significativa de comunicació entre els membres de la seva comunitat. Tenien un tipus d'escriptura jeroglífica on cada símbol podia tenir més d'un significat i les paraules s'escrivien com es pronunciaven, excepte les vocals, les quals s'ometien. Els fenicis, en canvi, ja tenien un alfabet molt més similar al que després van utilitzar els grecs, que van agregar les vocals. L'escriptura dels fenicis s'escrivia de dreta a esquerra, les paraules estaven separades entre elles i s'escrivia mitjançant línies entre mig. Va ser una evolució lenta.

Ja que la comunicació entre els éssers s'anava desenvolupant i creixent, amb el pas del temps, s'havia de poder crear un sistema que permetés comunicar-se d'un lloc a un altre. Per exemple, els egipcis van descobrir un material per a escriure. Aquest s'extreia del tall d'una planta anomenada papir d'on, temps més tard, es va crear el pergamí.

Però, la informació escrita tenia un risc, el descobriment del papir suposava la revelació de la informació. Les primeres civilitzacions de la història van desenvolupar tècniques per a poder enviar informació a través de missatges de manera que si el receptor de la informació era descobert la informació que portava no pogués caure en mans de l'enemic i ser descoberta. En aquest punt aparegué l'esteganografia juntament amb la criptografia. Bàsicament es diferencien en què quan parlem d'esteganografia amaguem la informació, en canvi amb criptografia la informació es visible però no es pot entendre si no es coneix el mecanisme de xifratge.

Així doncs en el segle II a. C. aparegué el xifrat de Polybios, aquest va ser el primer sistema de xifratge per substitució. Aquest sistema utilitzava la següent matriu donava a la figura 2.1. En aquest sistema, cada lletra quedava codificada per el parell de lletres corresponent a la fila/columna de la matriu. Tenia un problema: algunes lletres tenien la mateixa representació (col·lisions) però el context ajudava al desxifrar-les. D'altra banda el fet d'aplicar aquest mètode doblava la llargada del text, fet que consumia més papir.

	A	B	C	D	E
A	a	b	c	d	e
B	f	g	h	ij	k
C	l	m	nñ	o	p
D	q	r	s	t	u
E	v	w	x	y	z

Figura 2.1: Matriu de Polybios

Un altre mètode criptogràfic clàssic va ser el xifrat del César (segle I a. C.):

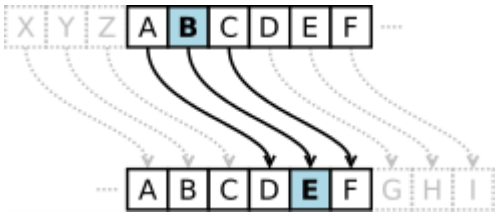

Figura 2.2: Xifrat en cèsar
Text original: ABCDEFGHIJKLMNÑOPQRSTUVWXYZ
Text codificat: GHIJKLMNOPQRSTÑUVWXYZABCDEF

Durant els segles següents es va seguir avançant amb el desenvolupament de diversos sistemes cada cop mes complexos. Aquests sistemes serien anomenats algoritmes de xifratge simètrics i mono alfabètics doncs la clau de desxifratge és la mateixa que la de xifratge i és única.

La totalitat d'aquests sistemes però va ser "trencada" a l'aparèixer la tècnica d'anàlisi de freqüència. Aquestes anàlisis utilitzen el coneixement de la llengua per a diferenciar quines lletres són més usades i en quin percentatge. Com més llarg és el text xifrat més senzill és descobrir per anàlisi de freqüències quin és el contingut original del text. Per a poder fer front a aquest sistema van aparèixer els xifratges polí alfabètics que permetien evitar l'anàlisi de freqüències.

Fins a la Segona Guerra Mundial no va haver grans avenços en el món de la criptografia. Va esser en la Segona Guerra Mundial on l'enginy i la mecanització van ser usades per a generar aparells de gran complexitat a fi de generar missatges impossibles de desxifrar. La més famosa va esser la màquina de codificació Enigma.

Figura 2.3: Codificador enigma

2.2 Algoritmes d'autentifiació

Les funcions hash són algoritmes matemàtics que transformen qualsevol bloc arbitrari de dades en un bloc de caràcters de longitud fixada, anomenat resum. El resum (hash) és el resultat de dita funció o algoritme. Alguns exemples són el MD5, el SHA-1 i en general els CRCs. Una propietat fonamental de les funcions d'autentificació (o resum) és precisament que són funcions resistens a col·lisions, és que si dos resums, utilitzant la mateixa funció, són diferents, llavors les dues entrades que generaren aquests resums també ho són. La funció resum, s'utilitza principalment tant en el camp de la criptografia com en el de la indexació de dades i en el de les comunicacions digitals com a codi de redundància per a corregir errors de transmissió. Molts sistemes relacionats amb la seguretat informàtica, amb les bases de dades o les transmissions de dades, usen funcions o taules de resum.

Normalment aquestes funcions han de tenir les següents característiques:

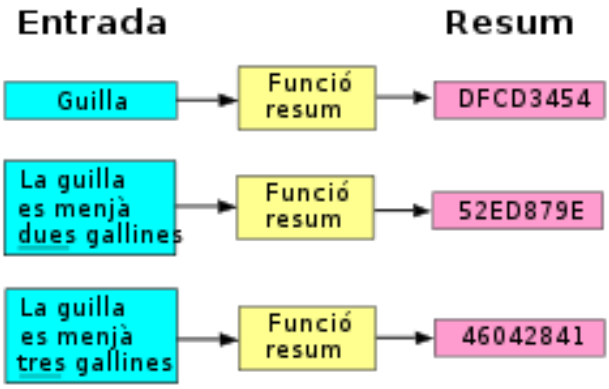

Figura 2.4: Funció resum (hash function)

1. Baix cost computacional i de memòria.
2. Compressió de dades.
3. Uniformitat : valors de sortida imparcialment distrubuïts.
4. Determinisme : amb el mateix valor d'entrada sempre s'ha d'obtenir la mateixa sortida.
5. Resistents a col·lisions

2.3 Algoritmes criptogràfics

En aquesta secció, es presentaran els principals algoritmes criptogràfics usats actualment, així com els algoritmes d'intercanvi de claus.

2.3.1 Intercanvi de claus Diffie-Hellman

W. Diffie i M.E.Hellman va proposar en el 1976 un sistema d'intercanvi de claus [4]. Aquest és un mètode específic d'intercanvi de claus criptogràfiques. És un dels primers exemples pràctics d'intercanvi de claus implementat dins del camp de la criptografia. El mètode d'intercanvi de claus Diffie-Hellman permet, que dues parts, les quals no tenen coneixement previ de l'altra, poder establir conjuntament una clau secreta compartida a través d'un canal de comunicació insegur. Aquesta clau pot ser utilitzatada per xifrar les comunicacions posteriors utilitzant un sistema de xifrat de clau simètrica.

Encara que l'intercanvi de claus Diffie-Hellman és un protocol clau-acord anònima (no autenticat), aquest proporciona la base per a una gran varietat de protocols autenticats, i s'empra per proporcionar confidencialitat directa perfecta en les capes de transport de seguretat. Aquest mètode va ser seguit poc després per RSA, una implementació de la criptografia de clau pública utilitzant algoritmes asimètrics.

2.3.2 Criptografia asimètrica o de clau pública

Suposem que el Bob vol enviar a l'Alícia un missatge secret que només ella pugui llegir. Alícia envia a Bob una caixa amb un pany oberta, de la qual només Alícia té la clau. Bob rep la caixa, escriu el missatge, el posa a la caixa i la tanca amb el seu pany (ara Bob no pot llegir el missatge). Bob envia la caixa a Alícia i ella l'obre amb la seva clau. En aquest exemple, la caixa amb el pany és la «clau pública» d'Alícia, i la clau del pany és la seva «clau privada».

La criptografia asimètrica o de clau pública, és una tipus de criptografia en la qual la clau utilitzada per xifrar un missatge difereix de la clau utilitzada per desxifrar-lo. En la criptografia de clau pública, un usuari té un parell de claus: una de pública i una de privada. La clau privada es manté en secret, mentre que la clau pública es pot distribuir lliurement. Els missatges s'han de xifrar amb la clau pública del receptor i aquests només podran ser desxifrar amb la seva clau privada corresponent. Les claus es relacionen matemàticament, però la clau privada a la pràctica no es pot obtenir a partir de la clau pública, o almenys d'una forma senzilla.

El criptosistema RSA

En criptografia, RSA (Rivest, Shamir i Adleman) és un sistema criptogràfic de clau pública desenvolupat en 1977 [5]. És el primer i més utilitzat algorisme d'aquest tipus i és vàlid tant per xifrar com per a signar digitalment. La seguretat d'aquest algorisme radica en el problema de la factorització de nombres enters. Els missatges enviats es representen mitjançant nombres. La seva seguretat es basa en la dificultat de factoritzar el producte, conegut, de dos nombres primers grans triats a l'atzar i mantinguts en secret. Actualment aquests primers són de l'ordre de 10^{200}, i es preveu que la seva mida creixi amb l'augment de la capacitat de càlcul dels ordinadors.

Tècnicament, en Bob envia a l'Alícia un «missatge pla» M en forma d'un nombre m menor que un altre nombre n, mitjançant un protocol

reversible conegut com *padding scheme* («model d'emplenament»).
A continuació genera el «missatge xifrat» c mitjançant l'operació:

$$c \equiv m^e \pmod{n},$$

on e és la clau pública d'Alícia.
Ara Alícia desxifra el missatge en clau c mitjançant l'operació inversa donada per

$$m \equiv c^d \pmod{n},$$

on d és la clau privada que només Alícia coneix.

El criptosistema ElGamal

Aquest sistema de xifrat basa la seva seguretat en el problema del logaritme discret (DLP), computacionalment difícil al no ser coneguts algoritmes eficients per a calcularlo.
Va ser descrit per Taher Elgamal en 1984 [6] i s'usa en programari GNU Privacy Guard, versions recents de PGP, i altres sistemes criptogràfics. Aquest algorisme no està sota cap patent el que ho fa d'ús lliure.
El procediment de xifrat (i desxifrat) està basat en càlculs sobre un grup cíclic finit G, que usualment és el grup $\mathbb{Z}_p^*$ amb l'operació producte. Més concretament, s'agafen com a paràmetres: g un generador de G, x un enter (clau privada) i $h = g^x$ (clau pública). Aleshores per xifrar el missatge m s'escull un aleatori r i calculem

$$(c_1, c_2) = (g^r, m \cdot h^r)$$

Per desxifrar amb la clau privada fem

$$m = \frac{c_2}{c_1^x}$$

L'algorisme d'ElGamal també pot ser utilitzat tant per a generar signatures digitals.

2.3.3 Criptografia amb corbes el·líptiques

En les últimes dècades, la criptografia amb corbes el·líptiques (ECC) ha adquirit una creixent importància. Aquesta criptografia també està basada

en problema del logaritme discret però enlloc de traballar sobre el grup $\mathbb{Z}_p^*$, treballant sobre el grup de punts d'una corba el·líptica (ECDLP). La seva característica principal ha estat aconseguir el mateix nivell de seguretat que RSA amb longituds de clau molt més curtes.

DLP & RSA (bits)	ECDLP (bits)	Ratio tamaño claves	AES (bits)
1024	160	1:6	
2048	224	1:9	
3072	256	1:12	128
7680	384	1:20	192
15360	512	1:30	256

Figura 2.5: Taula comparativa entre longitud/seguretat entre RSA, ECDLP i AES

Capítol 3

Fonaments del Bitcoin i de la Blockchain

L'any 1996, en Milton Fridman (Premi Nobel en Econòmiques i Finances) va comunicar en una entrevista "The one thing that is missing, but that soon will be developed, is a reliable e-cash, a method whereby on the Internet you can transfer fund from A to B without A knowing B or B knowing A"(http://youtu.be/mlwxdyLnMXM). Finalment aquest temps va ser 13 anys. Milton va morir 3 anys abans que algú plantejés sobre paper i realitzés una implementació.

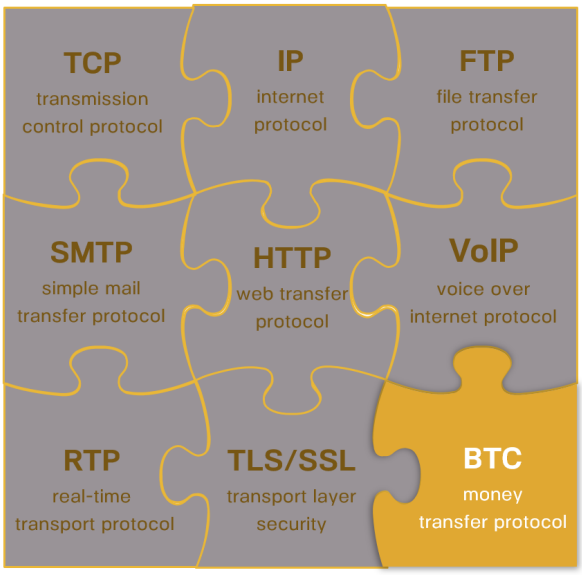

Figura 3.1: Protocols d'Internet

3.1 Protocol P2P

Les xarxes d'igual a igual [8] (peer to peer o P2P en anglès), són un sistema de comunicació per Internet que no té clients ni servidors fixos, sinó una sèrie de nodes que es comporten a l'hora com a clients i com a servidors dels altres nodes de la xarxa, en el qual les dades o les metadades es transfereixen a través d'una xarxa dinàmica. Aquesta tecnologia permet l'intercanvi de fitxers en xarxa sense que calgui un servidor central que redistribueixi les dades.

Aquest model contrasta amb el model client-servidor: qualsevol node pot iniciar o completar una transacció compatible, convertint doncs els PCs en elements actius que permeten als usuaris intercanviar informació i/o agrupar capacitat de processament, fent-ho a més al marge de la World Wide Web i sense necessitar un servidor centralitzat.

Història
La idea de compartir arxius d'igual a igual va començar a la dècada del 1960 quan es va crear la xarxa ARPANET (antecessor d'Internet). ARPANET fou creat per compartir arxius entre centres d'investigació dels Estats Units d'Amèrica i cada node de la xarxa tenia la mateixa importància.

Les primeres xarxes P2P [8] que sorgiren van ser les híbrides o centralitzades, utilitzant servidors per a l'obtenció de metadades (informació que indexa les dades o els recursos interns a la xarxa). Les principals funcions que s'implementaven en el servidor centralitzat d'una xarxa de compartició de fitxers eren totes les necessàries per posar en contacte el node buscador i el posseïdor.

Tot i així, els sistemes centralitzats de compartició de fitxers van assolir un èxit massiu a finals de l'any 1999 gràcies a Napster ideat per Shawn Fanning. Aquest va ser el primer sistema de distribució d'arxius que permetia als seus usuaris compartir les seves col·leccions de fitxers mp3 fàcilment. Aquest fet portà Napster als jutjats, ja que diferents discogràfiques veien perillar el seu negoci. Tot això va comportar un increment de la popularitat de la xarxa fins a assolir un màxim de 13,6 milions d'usuaris als inicis del 2001.

Aproximadament al cap d'un any del naixement de Napster, es va presentar la xarxa eDonkey2000 (ed2k) que superava en molts aspectes la seva predecessora. El fet de ser concebuda com a xarxa de compartició de qualsevol tipus d'arxius sense limitar-se a fitxers mp3, va fer que nasqués amb característiques més evolucionades com el suport a la descàrrega simultània de diferents porcions d'un sol arxiu provinents de diversos clients. D'aquesta

manera s'explotava l'amplada de banda de tots els clients sense limitar-se a un sol peer. Altres millores provenien de la utilització de funcions hash per identificar arxius o de la recerca interna entre servidors, per localitzar fitxers en clients llunyans.

Actualment s'està treballant per eliminar aquest inconvenient amb les xarxes P2P de tercera generació, les xarxes estructurades (structured peer-to-peer overlays), com és el cas de CAN, Chord, Pastry, Tapestry i que neixen amb la intenció de proporcionar un substrat per a aplicacions d'alta disponibilitat i escalabilitat, i a la vegada complir el paradigma del P2P (oferir un sistema capaç de funcionar d'una forma autònoma sense requerir una entitat centralitzada).

Funcionament
S'utilitza fonamentalment per compartir arxius per internet: textos, arxius audiovisuals, imatges i d'altres recursos informàtics de tota mena, els quals es distribueixen la càrrega de forma apropiada entre els nodes. Això permet la distribució massiva de fitxers, en perfecta escalabilitat, sense necessitat d'un servidor central. Per a això els destinataris finals són a l'hora nodes que col·laboren en els metamissatges de la xarxa i reenvien les dades que reben a altres interessats.

3.2 Blockchain

Com hem comentat un dels grans usos de la criptografia pot ser el de verificar identitats. En la vida Offline, tant en transaccions bancàries, com en accions legals sempre ha estat necessari l'ús d'una tercera part únicament per verificar la identitat. Els bancs o notaris per exemple compleixen aquesta funció. Ja en el món digital, hi ha desenvolupats certificats digitals o el DNI Electrònic que treballant amb conceptes de clau pública/privada i es concideren confiables per a realitzar un gran nombre d'activitats online. Tot i l'ús d'aquests certificats, continua essent necessari l'ús de tercers. Empreses com Paypal o el govern central són els verificadors en aquests casos. En aquest punt doncs ens sorgeix una necessitat, què podem fer per a que no siguin necessaris aquests intermediaris? Si analitzem aquests tercers amb deteniment podem observar que aquestes entitats són "confiables", alguna de les següents premisses:

1. Tenen tota la informació: Per exemple confiem en vendre el cotxe a un altre ciutadà ja que el banc ens acredita que aquest té els bens suficients

per comprar-lo doncs el banc te aquesta informació. Nosaltres no sabem si aquest altre ciutadà té 100 o 100000 unitats de moneda però sabem que són suficients per a vendre-li el cotxe. El banc doncs en aquest punt té aquesta informació.

2. Tenen una entitat superior que pot tenir tota la informació: El mateix exemple anterior es podria aplicar si nosaltres i l'altre individu treballen amb entitats bancàries diferents. Aquestes poden preguntar entre elles per a disposar de la informació. Amb l'exemple del notari seria similar: un notari per exemple ens acceptaria legalment a l'hora de firmar una hipoteca l'ús del DNI, la nostra firma... sense disposar dels mecanismes necessaris per a verificar la identitat al 100% doncs tant el DNI com la firma podrien ser falses però podria escalar la consulta per exemple al govern central o a un expert en firmes per a verificar-ho.

Així doncs el concepte més important és la informació ja que es la que dona la certesa. Així doncs, què pasaria si tothom tingués la mateixa informació? Aquest és el concepte de cadena de blocs o Blockchain.

La Blockchain [10] és doncs una base de dades en la qual tots els que participin tenen o poden tenir una còpia. A diferència de l'exemple del banc on el banc té la informació i sol ens verifica si l'altre individu pot o no pagar, en la Blockchain tots disposen de la informació. Aquesta base de dades conté el llibre de registre de totes les transaccions que s'han realizat. Cada participant és anomenat node i es conecta amb altres nodes de forma descentralitzada sense que cap node tingui més importància que els altres. Aquestes xarxes son anomenades P2P (peer to peer).

El missatge que transmeten s'anomena token. Un token no és més que una representació de la informació que emmagatzema la xarxa. Aquesta informació pot contenir qualsevol tipus de informació (transferències bancaries, fitxers ...) i va xifrada d'extrem a extrem entre els nodes.

Aquestes transferències de token a la vegada poden agrupar-se en blocs que es van generant cada cert temps. Les noves transferències de token que no han entrat al bloc anterior entrarien al següent bloc. Aquests blocs a la vegada estan referenciats entre ells i així successivament. Veure figura 3.2. D'aquí el nom de Blockchain.

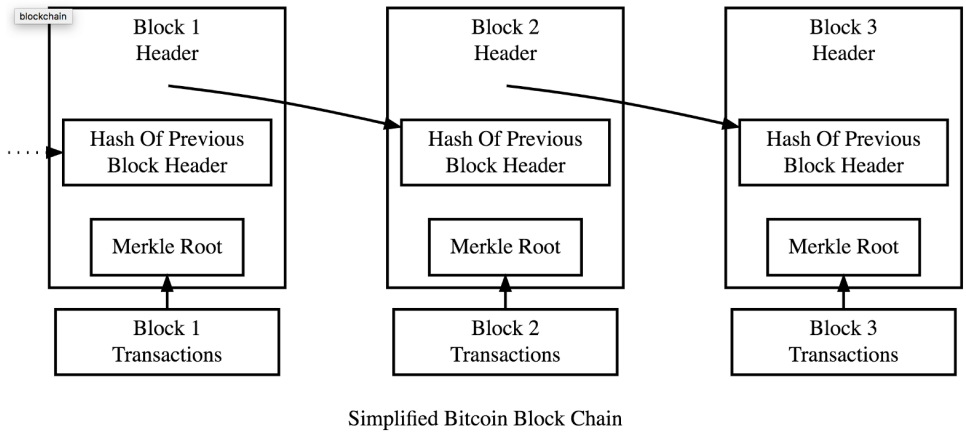

Figura 3.2: Blockchain

Aplicacions

El concepte de Blockchain o cadena de blocs és usat principalment en els següents camps:

1. En el camp de les criptomonedes la cadena de blocs s'usa com a notari públic confiable de tot el sistema de transaccions, a fi d'evitar el problema que una moneda es pugui gastar dues vegades. Per exemple, és usada en Bitcoin, Ethereum, Dogecoin i Litecoin, encara que cadascuna amb les seves particularitats.

2. En el camp de la certificació i firma digital s'usa la blockchain a fi d'emmagatzemar i donar validesa a documentació oficinal. Un exemple seria Certiblock. Certiblock es una blockchain basada sobre Ethereum que té com a finalitat acreditar les titulacions universitàries de diverses universitats Espanyoles.

3. En el camp de la proba d'existència, s'usen els atributs de persistència i d'integritat a fi de marcar en la Blochain un document o un identificador d'aquest amb una marca de temps que assegura l'existència en un moment determinat.

4. També és utilitzat com a base de plataformes descentralitzades que permeten suportar la creació d'acords de contracte intel·ligent (Smart-Contracts) entre parells. Exemples d'aquest tipus de plataformes són Ethereum.

Classificació

Les cadenes de blocs es poden classificar basant-se en l'accés a les dades emmagatzemats en la mateixa:

1. Cadena de blocs pública: és aquella en la qual no hi ha restriccions per llegir i escriure dades.

2. Cadena de blocs privada: és aquella en la qual hi ha una llista de entitats que tenen permisos per llegir i/o escriure dades

Per altra banda les cadenes de blocs es poden classificar basant-se en els permisos per a generar blocs en la mateixa :

- Cadena de blocs sense permisos: és aquella en la qual no hi ha restriccions perquè les entitats puguin processar transaccions i crear blocs. Aquest tipus de cadenes de blocs necessiten tokens per a proveir incentius que els usuaris mantinguin el sistema. Exemples de tokens són els nous bitcoins que s'obtenen en construir un bloc i les comissions de les transaccions.

- Cadena de blocs amb permisos: és aquella en la qual el processament de transaccions està desenvolupat per una llista predefinida de subjectes amb identitats conegudes.

Les cadenes de blocs públiques poden ser sense permisos (ex. Bitcoin) o amb permisos (ex. cadenes federades). Les cadenes de blocs privades han de ser amb permisos. Les cadenes de blocs amb permisos no han de ser privades ja que hi ha diferents formes d'accedir a les dades de la cadena de blocs, com per exemple:

- Llegir les transaccions de la cadena de blocs, potser amb algunes restriccions (exemple un usuari pot tenir accés només a les transaccions en les quals està involucrat directament).

- Proposar noves transaccions per a la inclusió en la cadena de blocs.

- Crear nous blocs de transaccions i afegir-ho a la cadena de blocs.

3.2.1 Conceptes màtematics de la blockchain

La blockcain de Bitcoin, que usarem en el següent capítol per a la implementació de la nostra prova d'existència, utilitza l'algoritme de signatura digital ECDSA (Elliptic Curve Digital Signature Algorithm), amb el que es creen les parelles de claus (pública/privada) que permeten realitzar transaccions amb Bitcoins. Aquest algoritme és una variant del DSA (Digital Signature Algorithm) basada en el problema del logaritme discret sobre corbes el·líptiques (ECC).

Com hem vist en el capítol de conceptes bàsics de Criptografia, el principal avantatge d'usar criptografia amb corbes el·líptiques és que amb una longitud de clau més curta s'assoleixen valors igual de segurs que amb ElGamal o RSA. Això per la implementació de la Blockchain de Bitcoin es torna important doncs la clau ha de ser fàcilment transferible mitjançant codis QR, o bé ser impresa.

L'algoritme ECDSA crea claus de 256 Bits de longitud codificats en base 58, que dona com a resultat claus de 44 caràcters. Una clau per al DSA necessitaria de 350 dígits per assolir el mateix nivell de seguretat. Així doncs el principal motiu d'utilitzar ECC era facilitar l'ús de les direccions públiques del protocol Bitcoin.

Tot i així Satoshi Nakamoto va decidir que 44 caràcters eren massa per una direcció pública va decidir aplicar un procés de funcions hash per la creació de les Claus públiques que és el següent:

1. Disposar de la parella de claus ECDSA Pública i Privada
2. Aplicar la funció de Hash SHA-256 de la clau Pública
3. Sobre aquest resultat, aplicar la funció de Hash RIPEMD-160
4. Agregar al davant del resultat un Byte que correspon a la versió de la Blockchain, per defecte 00
5. Aplicar de nou la funció SHA-256 sobre aquest resultat
6. Aplicar de nou la funció SHA-256 sobre aquest resultat
7. Els primers 4 Bytes d'aquesta direcció resultant seran la suma de comprovació
8. Agregar aquests 4 Bytes al final del resultat de l'etapa 4
9. Convertir aquest resultat a base 58. El resultat tindrà entre 27 i 34 dígits

3.2.2 Prova d'existència sobre blockchain

Es defineix prova d'existència com un procediment que es pot aplicar a un determinat fitxer i ens permet saber si aquest fitxer existia en un moment

determinat del temps amb el mateix contingut que en l'actualitat. Així doncs aplicant els coneixements explicats previament podem assolir l'objectiu.

L'objectiu d'aquest treball de final de màster és realitzar la nostra implementació de prova d'existència sobre la Blockchain de Bitcoin i la funció de Hash MD5.

Capítol 4

Implementació i prova de concepte(d'existència)

En aquest capítol explicarem com em realitzat la nostra implementació de prova d'existència sobre la Blockchain de Bitcoin. Des de la creació d'un node fins al desenvolupament de una petita WebApp de prova d'existència.

4.1 Creació d'un node Blockchain

Per a la instal·lació d'un node de la Blockchain de Bitcoin [12] he usat la distribució Ubuntu per la seva estabilitat i facilitat d'instal·lació dels paquets necessaris per la creació del Node.

- 200Gb de Espai Lliure doncs la Blockchain a (01/2018) és d'unes 150Gb de dades.
- 2Gb de Ram
- Conexió a internet 1 Mb/s simètric amb el port 8333 TCP publicat.

Els passos a seguir per a la instalació són els següents:
apt-get install software-properties-common

Afegim el repositori de bitcoin
sudo apt-add-repository ppa:bitcoin/bitcoin

Actualitzem la BD dels repositoris locals
sudo apt-get update

Instal·lem el Bitcoin Core Daemon i la interfície gràfica (no es necessària).
sudo apt-get install bitcoin-qt bitcoind

Una vegada instal·lat posem el servidor en marxa amb la sentència *bitcoind -daemon*

La interfície gràfica ens permet utilitzar-la com a moneder així com fer i rebre transferències de forma senzilla. Per a començar a operar amb el nostre node de bitcoin hem d'esperar unes hores fins que la nostra BD estigui actualitzada.
Podem validar la instal·lació i l'estat de la BD amb les comandes:
bitcoin-cli getconnectioncount

Aquesta ens indica el nombre de connexions establertes al nostre node.
bitcoin-cli getblockcount
L'anterior ens indica el nombre de blocs que consta la nostra còpia de la Blockchain i hauria de coincidir amb el valor d'una web com per exemple: https://blockchain.info/es
Al cap d'unes hores el nostre node Bitcoin ja formarà part de la Blockchain mundial. Podem veure l'estat de la Blockchain a: https://bitnodes.earn.com/

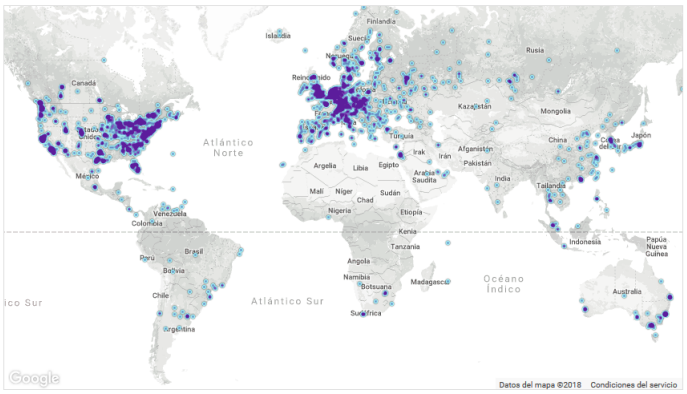

Figura 4.1: Nodes a nivell mundial

Aquí podem veure el nostre node ubicat a la IP: 5.40.179.133 que correspon al CPD que tenim a Lleida, lloc on s'ubica el nostre node.

Figura 4.2: Node local

4.2 Treballar amb la Blockchain

Mitjançant l'eina bitcoin-cli

Comanda	Utilització
bitcoin-cli listaccounts	Llistar els comptes de bitcoin amb les seves quantitats
bitcoin-cli getnewadress	Obtenir una nova adreça de moneder
bitcoin-cli gettransaction {*transacctionid*}	Obtenir informació sobre una transacció
bitcoin-cli getadressesby account {*account*}	Obtenir les adreces corresponent a un compte
bitcoin-cli getbalance {*account*}	Mostrar el balanç actual d'un compte
bitcoin-cli dumpprivkey {*walletid*}	Fa un 'dump' de la clau privada d'un moneder

Per escriure a la Blockchain de Bitcoin utilitzarem el camp OP_RETUN. Aquesta operació està implementada des de la versió de Bitcoin Core 0.9.0. Inicialment es disposaven de 40 bytes d'informació, posteriorment en una revisió va augmentar a 80 bytes. En el nostre cas usarem una de les moltes llibreries que hi ha per escriure a la blockchain usant el camp OP_RETURN per a PHP de:

https://github.com/coinspark

En aquest punt emplenarem les dades del fitxer general de configuració OP_RETURN.php

Algorithm 1 Configuració OP_RETURN.php

```
define('OP\_RETURN\_BITCOIN\_IP', '127.0.0.1'); // IP address of your bitcoin node
define('OP\_RETURN\_BITCOIN\_USE\_CMD', false); // use command-line instead of JSON-RPC?

if (OP\_RETURN\_BITCOIN\_USE\_CMD) {
  define('OP\_RETURN\_BITCOIN\_PATH', '/usr/bin/bitcoin-cli'); // path to bitcoin-cli executable on this server
}  else {
  define('OP\_RETURN\_BITCOIN\_PORT', ''); // leave empty to use default port for mainnet/testnet
  define('OP\_RETURN\_BITCOIN\_USER', ''); // leave empty to read from ~/.bitcoin/bitcoin.conf (Unix only)
  define('OP\_RETURN\_BITCOIN\_PASSWORD', ''); // leave empty to read from ~/.bitcoin/bitcoin.conf (Unix only)
}

define('OP\_RETURN\_BTC\_FEE', 0.0001); // BTC fee to pay per transaction
define('OP\_RETURN\_BTC\_DUST', 0.00001); // omit BTC outputs smaller than this
define('OP\_RETURN\_MAX\_BYTES', 80); // maximum bytes in an OP\_RETURN (80 as of Bitcoin 0.11)
define('OP\_RETURN\_MAX\_BLOCKS', 10); // maximum number of blocks to try when retrieving data
define('OP\_RETURN\_NET\_TIMEOUT\_CONNECT', 5); // how long to time out when connecting to bitcoin node
define('OP\_RETURN\_NET\_TIMEOUT\_RECEIVE', 10); // how long to time out retrieving data from bitcoin node


```

Ara ja podem realitzar la primera prova d'escriptura utlitzant la següent comanda:
php send-{nom de l'arixiu de configuració}.php {adreça de destí}{quantitat}[comentari]

Un cop emplenem els paràmetres ens quedarà:

php send-OP_RETURN.php 16tzhZ1E5ZVHLXXC7pmQ1smb8E7oeUD5VZ 0.00001 'TEST TFM 3R'

Aquesta sentència ens retornarà el ID de la Transacció:

TxID: ccc239600a33290cf0f730777c2d1b72d349fbaf4ad0991753a88d2d766d9953

Podem veure la nostra transcció amb el seu ID utilitzant la web de visualització:
https://blockchain.info/tx/bd485efcf65be2871ad4c67602f1369d9d6f74f630d62ebfe85825caf5bfe19e

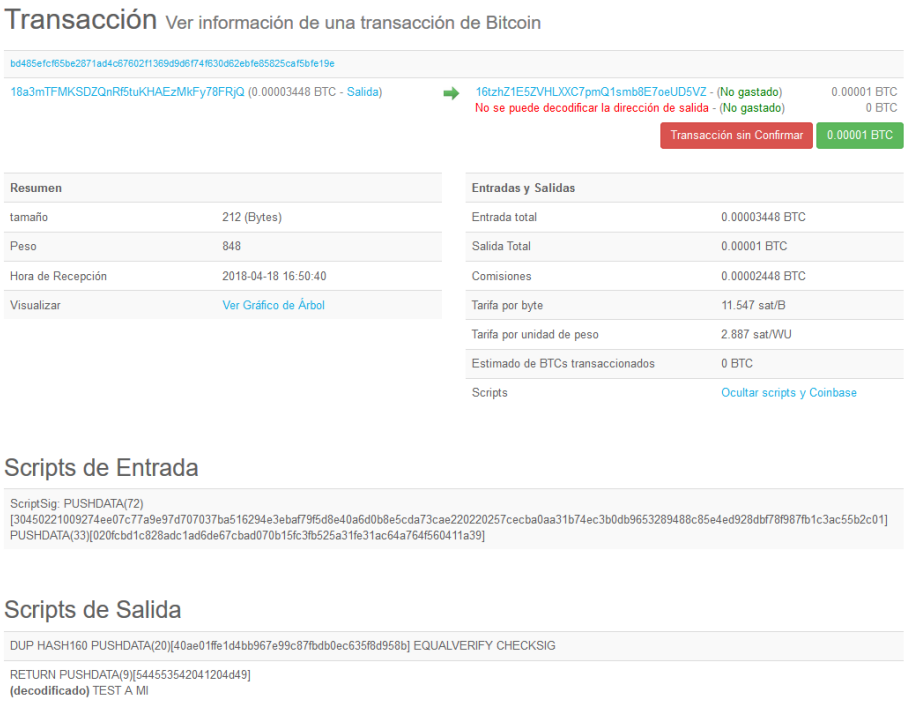

Figura 4.3: Transacció a blockchain.info

A l'apartat de Scripts de Sortida podem veure la descodificació del text enviat al camp OP_RETURN.

4.3 Creació d'una Webapp de prova d'existència

L'objectiu doncs era la creació de una aplicació Web. Aquesta aplicació web permet pujar un document, extreure'n el seu MD5 i guardar-lo com al camp OP_RETURN [13] a la Blockchain de Bitcoin. Una vegada el fitxer és pujat la app s'emmagatzema la MD5. Si el mateix fitxer es pujat una altra vegada i el MD5 concorda amb algun MD5 de la base de dades la WebApp ens retorna l'enllaç de la Blockchain que conté el contingut del MD5, juntament amb la marca de temps de la creació d'aquell registre. Això ens permet parlar de una prova de existència.

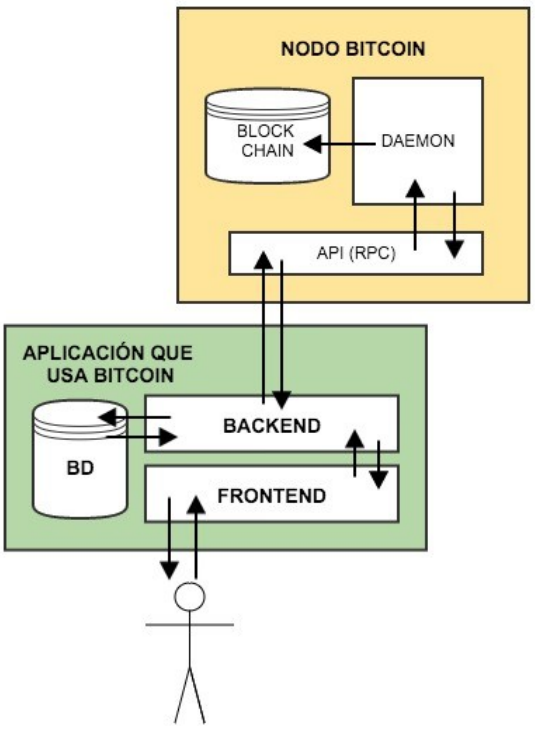

Figura 4.4: Esquema de funcionament del WebApp

Per a utilitzar la web app podem accedir a: guardo.vunkers.online

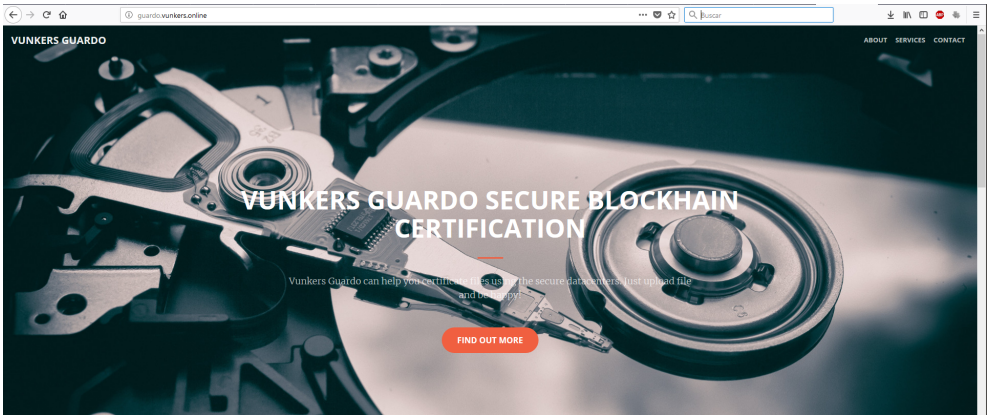

Figura 4.5: Pas 1: Pagina Principal de la nostra WebApp

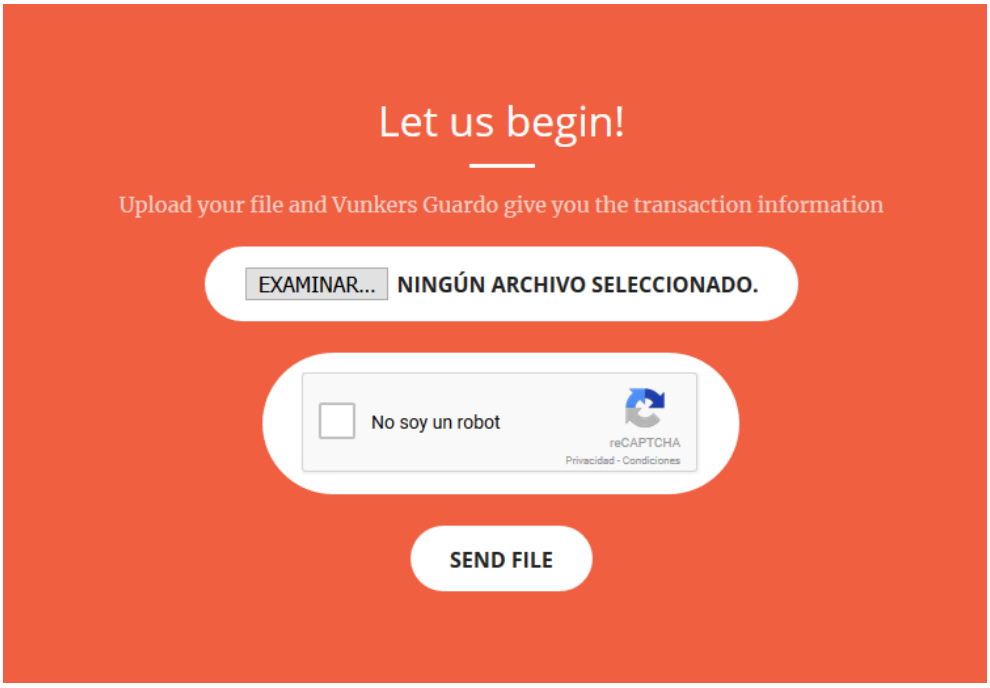

Figura 4.6: Pas 2: Apartat de la nostra WebApp que ens permet seleccionar el fitxer sobre el que volem crear o visualitzar marca de temps

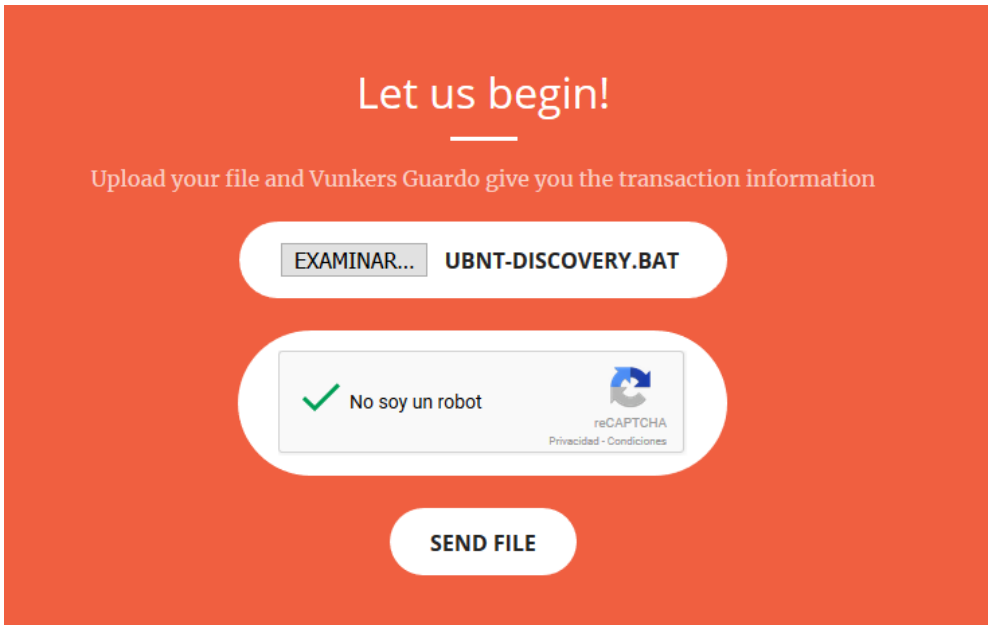

Figura 4.7: Pas 3: Una vegada tenim el fitxer seleccionat, apliquem el check de validació de RecaptCha

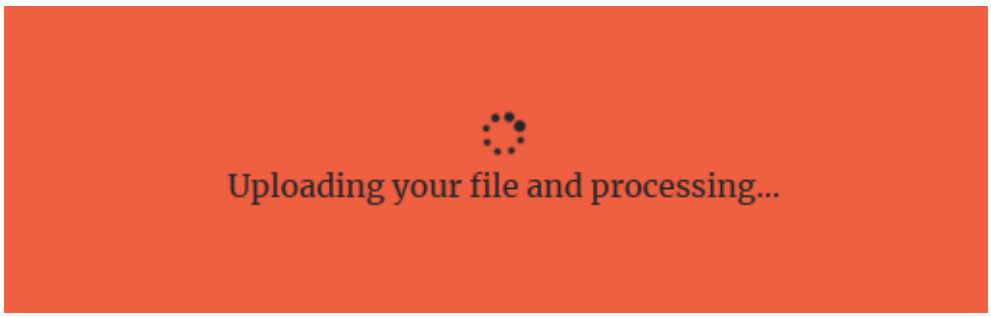

Figura 4.8: Pas 4: En aquest punt l'aplicació extreu el MD5 del fitxer pujat i el puja a la Blockchain de BTC amb un import molt petit i amb el MD5 del fitxer en el camp OP RETURN

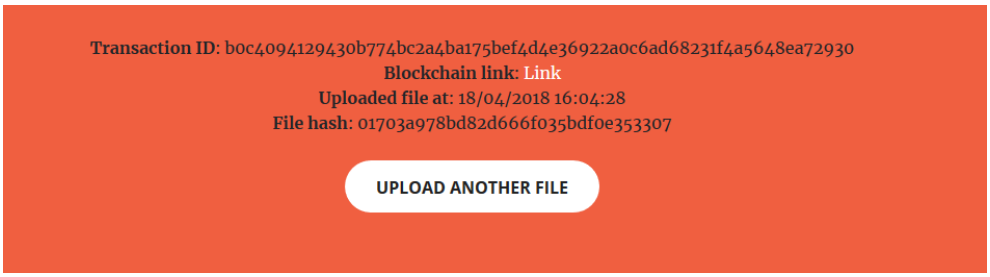

Figura 4.9: Pas 5: En aquesta captura observem que la WebApp retorna el numero de transacció en la Blockchain de BTC. Un enllaç per a que vuguem veure la nostra transacció en blockchain.com, la marca de temps i el MD5 del Fitxer.

Si es torna a pujar un fitxer que ja ha estat pujat abans, el sistema retorna la consulta ja realitzada anteriorment.

Capítol 5

Conclusions

Durant l'execució d'aquest TFM, hem hagut d'anar superant petits entrebancs a fi de dur a terme el nostre objectiu. Finalment va resultar més senzill de l'esperat doncs actualment hi ha moltes llibreries en diferents llenguatges que ens faciliten l'escriptura a la Blockchain de Bitcoin. Per altra banda, si bé de forma molt senzilla, és perfectament viable la implementació d'una prova d'existància sobre la cadena de blocs de Bitcoin no és la millor elecció per a realitzar-ho.

Actualment existeixen cadenes de blocs més avançades que implementen nativament més funcionalitats, no orientades a un sistema monetari sinó de contractes intel·ligents que serien més adients.

Com a projectes futurs es podria realitzar la mateixa aplicació sobre la cadena de blocs d'Ethereum.

Bibliografia

[1] Wikipedia The Free Encyclopedia (21-9-2017)
https://en.wikipedia.org/wiki/Cryptography

[2] *Josep M. Miret, Javier Valera, Magda Valls* CCE: Criptografía con curvas elípticas (14-03-2019)
http://www.criptored.upm.es/crypt4you/temas/ECC/leccion1/leccion1.html

[3] Logaritme discret: Wikipedia The Free Encyclopedia (22-2-2019)
https://ca.wikipedia.org/wiki/Logaritme_discret

[4] *Whitfield Diffie, Martin Hellman* New directions in Cryptography. IEEE Transactions on Information Theory, vol. IT-22, 1976, 644-654.

[5] *Ron Rivest, Adi Shamir, Leonard Adleman* A method for obtaining digital signatures and PKC. Communications of the ACM. Vol. 21 (2), 1978, 120-128.

[6] *Taher Elgamal* A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Trans. Inform. Theory*, 31, no. 4, 469-472, 1985.

[7] Hash: Wikipedia The Free Encyclopedia (8-10-2017)
https://en.wikipedia.org/wiki/Hash_function

[8] P2P: Wikipedia The Free Encyclopedia (8-10-2017)
https://en.wikipedia.org/wiki/Peer-to-peer

[9] Especificacions Blockchain de Bitcoin: Bitcoin.org (20-4-2018)
https://bitcoin.org/bitcoin.pdf

[10] Blockchain: Wikipedia The Free Encyclopedia (23-4-2018)
https://en.wikipedia.org/wiki/Blockchain

[11] Technical background of version 1 Bitcoin addresses (6-5-2019)
https://en.bitcoin.it/wiki/Technical_background_of_version_1_Bitcoin_addresses

[12] Creació Node Bitcoin: Bitcoin.org (28-4-2018)
`https://bitcoin.org/en/full-node`

[13] Bitcoin OP RETURN: Bitcoin.org (28-4-2018)
`https://bitcoin.org/en/developer-guide`

[14] Guía de Estilo para lenguaje PHP: (23-4-2018)
`https://coppeldev.github.io/php/standards/psr-2.html`

[15] Nikos Drakos *Manual de Latex*. 1995, Computer Based Learning Unit